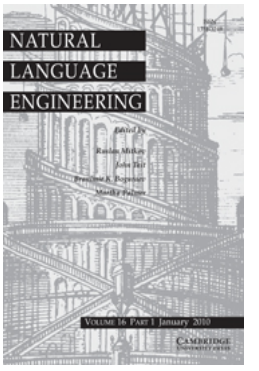
# Learning human multimodal dialogue strategies




# Learning human multimodal dialogue strategies


V. RIESER and O. LEMON



Natural Language Engineering / Volume 16 / Issue 01 / January 2010, pp 3 - 23
DOI: 10.1017/S1351324909005099, Published online: 22 April 2009

*Natural Language Engineering* **16** (1): 3–23. © Cambridge University Press 2009
doi:10.1017/S1351324909005099


# *Learning human multimodal dialogue strategies*

V. RIESER
*School of Informatics, University of Edinburgh, Edinburgh, EH9 8AB, GB*
*e-mail: vrieser@inf.ed.ac.uk*

O. LEMON
*School of Informatics, University of Edinburgh, Edinburgh, EH9 8AB, GB*
*e-mail: olemon@inf.ed.ac.uk*

(*Received July 2007; revised 7 January 2009; accepted 25 February 2009; first published online 22 April 2009*)

**Abstract**

We investigate the use of different machine learning methods in combination with feature selection techniques to explore human multimodal dialogue strategies and the use of those strategies for automated dialogue systems. We learn policies from data collected in a Wizard-of-Oz study where different human 'wizards' decide whether to ask a clarification request in a multimodal manner or else to use speech alone. We first describe the data collection, the coding scheme and annotated corpus, and the validation of the multimodal annotations. We then show that there is a uniform multimodal dialogue strategy across wizards, which is based on multiple features in the dialogue context. These are generic features, available at runtime, which can be implemented in dialogue systems. Our prediction models (for human wizard behaviour) achieve a weighted f-score of 88.6 per cent (which is a 25.6 per cent improvement over the majority baseline). We interpret and discuss the learned strategy. We conclude that human wizard behaviour is not optimal for automatic dialogue systems, and argue for the use of automatic optimization methods, such as Reinforcement Learning. Throughout the investigation we also discuss the issues arising from using small initial Wizard-of-Oz data sets, and we show that feature engineering is an essential step when learning dialogue strategies from such limited data.

## 1 Introduction

When designing interfaces for Human Computer Interaction (HCI), and dialogue systems in particular, one is interested in how humans perform the task that is to be supported by the interface. In designing a multimodal dialogue system, we are especially interested what kind of strategies humans apply when put in the place of such a system. A typical method is therefore to conduct a 'Wizard-of-Oz' (WOZ) experiment where several humans (so-called 'wizards') serve as hidden operators while the user is left in the belief that s/he is interacting with a real system. In contrast to conventional WOZ trials we are not only interested in the users' behaviour, but also in the behaviour of our human wizards. In particular, we use this data to construct a model of human multimodal behaviour. This model allows us to describe what kinds of decisions humans make when confronted with

the choices that must be made by an automated system, e.g. how they perform when having multiple, context dependent decisions to make, how they deal with input noise, and how they act under time constraints. These kind of models can be used to gain insights about 'natural' behaviour, to investigate whether human behaviour indeed would be optimal for an automated system, and also to discover what might be improved when building dialogue systems.

However, data from WOZ trials is expensive to collect, and as a result only a limited amount of data is available to answer those complex questions. Furthermore, there are many potentially relevant features in multimodal interaction which might influence human behaviour.

In this paper, we investigate the use of machine learning (ML) methods to explore human multimodal dialogue strategies when there is some interpretation uncertainty about user utterances. We conduct a WOZ experiment for a multimodal in-car music player, where we investigate when human wizards decide to perform a clarification request (CR) in a multimodal or speech-only manner. We then use ML techniques to explore those strategies, and discuss their use in automated dialogue systems. We find that the strategies employed by human wizards are sub-optimal and argue for the use of automatic optimisation methods, such as Reinforcement Learning (RL). Throughout the investigation we also discuss the issues arising from using small initial WOZ data sets, and we show that feature engineering is an essential step when learning dialogue strategies from such limited data.

Note that by 'clarification request' we mean a dialogue action which is designed to reduce uncertainty or ambiguity about the user's goals. An example of spoken CR in our music player domain would be: 'A rock song by which artist?', and an example multimodal CR would be, while displaying a list on the screen: 'I'm displaying a list of rock artists on the screen. Is it one of these?'

In dialogue application domains with high-interpretation uncertainty, for example, caused by acoustic uncertainties from a speech recogniser, multimodal generation and input processing may lead to more robust interaction (Oviatt 2002) and reduced cognitive load (Oviatt, Coulston and Rebecca 2004). On the other hand, in some situations, especially in situations imposing high cognitive load on the user such as driving, verbal output may be preferred over the use of graphics (Salmen 2002). Thus, our hypothesis is that multimodal generation should follow a context- and user-dependent strategy. Previous work on Natural Language Generation for information presentation in spoken dialogue systems (e.g. for browsing lists of flights or restaurants) has shown that it is useful to adapt the output to user preferences, e.g. (Walker *et al.* 2004), and cognitive load (Winterboer *et al.* 2007).

The overall method and corresponding structure of the paper is as shown in Figure 1. We proceed as follows: in Section 2 we present the data collection in a multimodal WOZ experiment and describe how this setup was used to elicit human multimodal clarification requests. Section 3 describes the annotations for multimodal CRs and their validation. For exploring human multimodal clarification strategies we extract potential contexts from the WOZ corpus using Information State Update (ISU) based features (Lemon *et al.* 2005), as described in Section 4. We apply feature engineering methods such as discretising numeric features, and we use feature selection methods to further analyse the data. These techniques also help

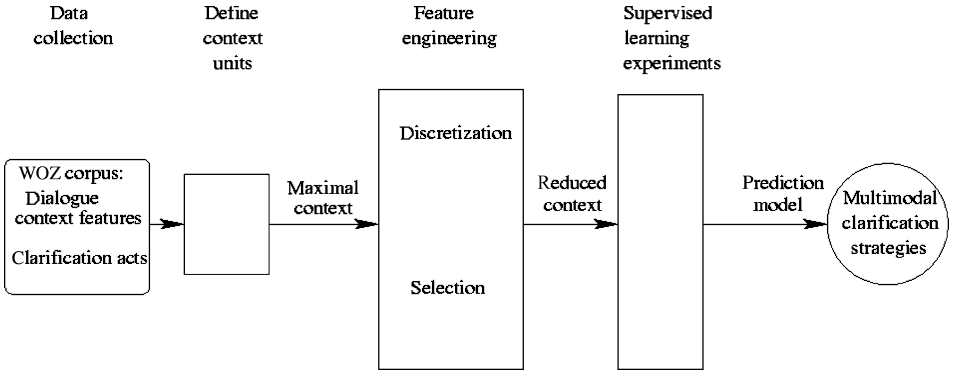



Fig. 1. Methodology and structure.

to reduce the context representation and thus the feature spaces used for learning from small amounts of data. In Section 5 we test different supervised classifiers upon these reduced contexts and separate out the independent contributions of learning algorithms and feature engineering techniques. We discuss and interpret the learned strategies, and their use in automated dialogue systems, and conclude in Section 6.

## 2 Data collection in a Wizard-of-Oz experiment

### *2.1 Motivation and goal of the experiment*

Previous work has investigated how humans ask for clarification in task-oriented dialogue (Rieser and Moore 2005). This work identified features influencing human clarification strategies (such as relation to task success, channel quality and modalities available). We now investigate how these findings transfer to multimodal human-machine interaction by collecting data on clarification strategies employed by multiple human wizards in a WOZ trial. We are especially interested in multimodal presentation strategies in situations where the wizard decides to clarify user utterances.

In the larger context of the TALK project[1] we developed an experimental setup to gather interactions where the wizard can combine spoken and visual feedback, namely, displaying results of a database search, and the user can both speak about or graphically select items on the screen. The corpus gathered with this setup is also known as the SAMMIE corpus (Kruijff-Korbayová *et al.* 2006a). The SAMMIE system provides an in-car multimodal conversational interface to a music player, see Figure 2. All the interactions are in German.

In contrast to conventional WOZ trials we were not only interested in the users' behaviour, but also in the behaviour of our human wizards. One goal of the WOZ experiment was to gather data on spoken and multimodal clarification strategies as employed by multiple human wizards and the performance of those strategies. In

[1] TALK (Talk and Look: Tools for Ambient Linguistic Knowledge; www.talk-project.org) was funded by the EU as project No. IST-507802 within the 6th Framework program.


Fig. 2. The TALK project's SAMMIE in-car music player dialogue system GUI.


Fig. 3. Example of display template choices for the wizards.

particular, we are interested what modality the wizards choose for CRs. Since we are interested in the range of possible human behaviours, the wizards' responses were not constrained by a script, but the wizard was able to talk freely and choose to show tables of retrieved items on the user's screen, where the graphical outputs were automatically generated via templates. An example of possible choices that the wizards could display is shown in Figure 3, where they can select whether to display albums, tracks or artists corresponding to a particular search. They could also opt to reply via voice alone. In this work we are interested in the binary problem whether the wizard would choose to show any screen output at all. We learn a model to predict when to generate a CR in a multimodal or speech-only manner.

In current work we investigate more complex generation strategies (see for example Rieser and Lemon 2009). Note that the amount and quality of the initial data limits the complexity of the learning problem. In Section 4 we apply feature engineering

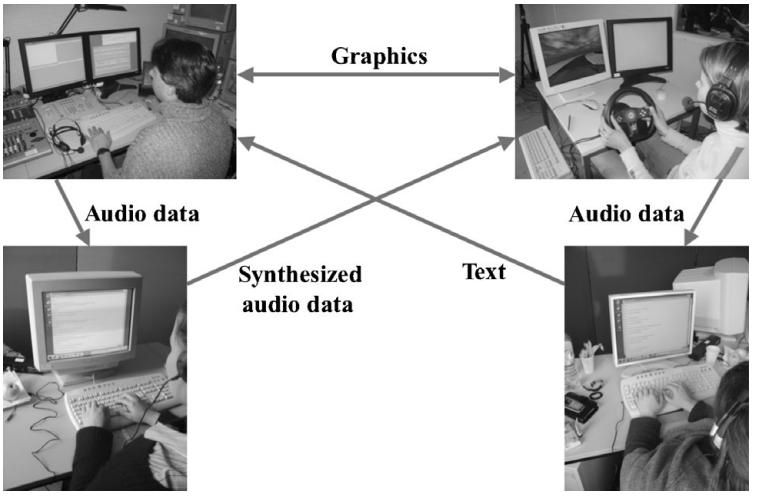



Fig. 4. Multimodal WOZ data collection setup for an in-car music player application, using the Lane Change driving simulator. Top right: user; top left: wizard; bottom: transcribers.

methods such as discretising numeric features, and we use feature selection methods in order to improve the quality of the data.

### *2.2 Experimental setup*

We now briefly summarise the details of the experiments. A full description of the setup can be found in Kruijff-Korbayová *et al.* 2005 and Rieser, Kruijff-Korbayová and Lemon (2005). The experimental setup is shown schematically in Figure 4. There are five people involved in each session of the experiment: an experiment leader (not shown), two transcribers, a user and a wizard.

The wizards play the role of an intelligent interface to an MP3 player and are given access to a database of music information. Subjects are given a set of predefined tasks and are told to accomplish them by using an MP3 player with a multimodal interface. In a part of the session the users also get a primary driving task, using the *Lane Change* driving simulator (Mattes 2003). This setup enabled the collection of dialogue data combining primary and secondary tasks in the experimental setup. However, it would be an interesting direction for future research to replicate the method described in this paper on data collected in a more realistic in-car environment.

The wizards can speak freely and display the search results or playlists on the screen. The users can also speak, as well as making selections on the screen. The user's utterances are immediately transcribed by a typist and also recorded. The transcription is then presented to the wizard. This was done in order to deprive the wizards of information encoded in the intonation of utterances (as in current dialogue systems), and in order to be able to corrupt the user input in a controlled way, simulating understanding problems at the acoustic level. The wizard's utterances are also transcribed (and recorded) and presented to the user via a speech synthesiser.

In total, the corpus contains data from 21 different subjects, who each participated in one session with one of our 6 wizards. Each subject worked on four tasks, first two without driving and then two with driving. The duration was restricted to twice 15 minutes. At the end of the experiment, users rate their satisfaction with the 'system'. Please see Rieser (2008) for further details. The tasks were to search for a song in the database, and to build a playlist satisfying certain constraints. We also made sure that none of the wizards attempted the same task twice, in order to avoid learning effects. The 21 experimental subjects were all native speakers of German with good English skills. They were all students (equally spread across subject areas), half of them male and half female, and most of them were between 20 and 30 years old.

### *2.3 Corpus description*

The SAMMIE corpus gathered with this setup comprises 21 sessions with 70 dialogues and approximately 17,000 turns. Our six wizards contributed about equal proportions to this data, i.e. about 12 dialogues each. Example 1 shows a typical multimodal clarification subdialogue,[2] concerning an uncertain reference (note that 'Nevermind' is both an album name by the band 'Nirvana' and a song title by the band 'The Red Hot Chili Peppers').

(1) **User:** Please play 'Nevermind'.
**Wizard:** Does this list contain the song? *[shows list with 20 DB matches]*
**User:** Yes. It's number 4. *[clicks on item 4]*

For each session we gathered logging information which consists of Open Agent Architecture (OAA) messages in chronological order, which contain various information, e.g. the transcriptions of the spoken utterances, the wizard's database queries and the numbers of results, etc. The data has been transcribed and automatically converted into NXT format (Carletta *et al.* 2003)[3] together with the information from the log files as described in Kruijff-Korbayová *et al.* (2006b).

### *2.4 Invoking clarification behaviour*

To approximate speech recognition errors we used a tool that 'deletes' parts of the transcribed utterances. Due to the fact that humans try to make sense of even heavily corrupted input, this method not only covers non-understandings, but wizards also built up their own hypotheses about what the user really said, which can lead to misunderstandings. We introduced different deletion rates, where the deletion rate is defined as the percentage of the total number of words in an utterance. Note that randomly deleting words in a very long utterance is less likely to cause understanding problems than randomly deleting words in a very short utterance. In future work we plan to refine this method and implement a semantic error rate, i.e. controlling for content words to be deleted. The word deletion rate varied: 20 per cent of the

[2] Translated from German.
[3] `http://www.ltg.ed.ac.uk/NITE/`.

utterances were weakly corrupted (= deletion rate of 20 per cent ), and 20 per cent were strongly corrupted (= deletion rate of 50 per cent ). In 60 per cent of the cases the wizard saw the transcribed speech uncorrupted. Example 2 illustrates the kind of corrupted utterances the wizard had to deal with.

(2) **Uncorrupted:** Zu dieser Liste bitte Track 'Tonight' hinzufügen.
[Add track 'Tonight' to this list.]
**Weak:** Zu dieser Liste bitte Track Tonight . . . .
[. . . track 'Tonight' to this list.]
**Strong:** Zu . . . Track Tonight . . . .
[. . . track 'Tonight' to . . . .]

There are some shortcomings of this technique, which are also pointed out by Schlangen and Fernandez (2007), who used a similar setup to simulate a noisy communication channel. First of all, deleting words is a rather crude simulation of real-world acoustic problems. Note that there are also studies introducing errors from ASR (Skantze 2005; Stuttle, Williams and Young 2004). We think, however, that deleting words is simulating more 'natural' communication problems (e.g. some parts of an utterance might be distorted by transient noise). It is not clear whether confronting human wizards with ASR errors will reveal the range of natural behaviour we are interested in. Furthermore, although the wizards were trained, they did not always use optimal strategies (as further discussed in Section 5.5). Last but not least, it is not always clear what kind of understanding problem caused the clarification. A commonly used method is to use the follow-up reply to the CR to identify the 'mutual agreed understanding' of the CR (Purver, Ginzburg and Healey 2003). This does not always work: for example, Rodriguez and Schlangen (2004) labelled 14.3 per cent cases as still being ambiguous. One of the reasons might be that the human subjects tend to 'over-answer', i.e. even though there was an acoustic problem they present an hypothesis on a higher level of understanding where the answer will resolve both potential problems. To circumvent this problem we directly ask the wizards to indicate the problem which caused the need for clarification. Every time the wizard asked a CR the experiment leader would invoke a pop-up window asking the wizard to indicate one of the possible sources as described below.

## 3 Annotation of multimodal clarification requests

### *3.1 Annotation scheme*

The data is annotated with the following annotation scheme for clarification requests, based on Rodriguez and Schlangen (2004). This scheme has been shown to be applicable for several different domains of dialogue (Rieser and Moore 2005), and thus supports clarification strategies which are portable. In this annotation scheme a clarification object is defined as a triple of three related utterances; the CR itself, the antecedent (i.e. the problematic user utterance which caused the CR), and the reply to that CR. For each of these three utterances we annotate additional attributes as shown in Figure 5.

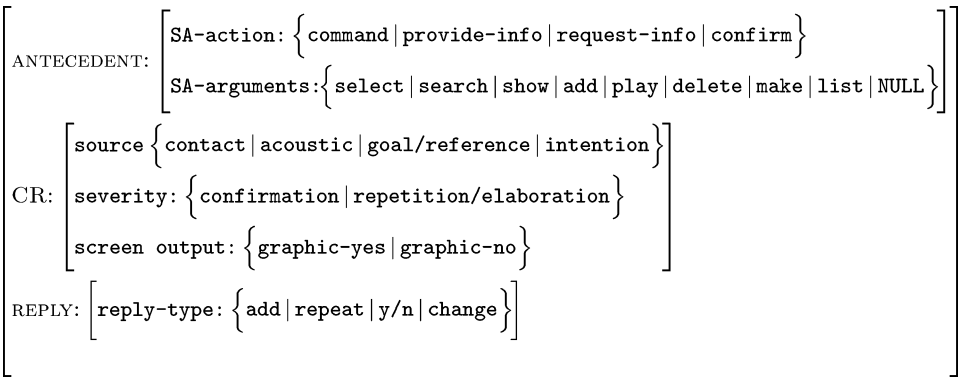



Fig. 5. The annotation scheme.

For the CR itself we manually annotate the degree of uncertainty (`severity`) as indicated by the speaker. The modality, i.e. whether the screen output was used (`graphic-yes`), or whether clarification was using speech only (`graphic-no`) was automatically annotated. The case that the wizard only used screen output for clarification did not occur. The problem `source` was already annotated online by the wizards and revised by the annotators. For some cases the annotators corrected the wizard's choice.[4]

Example 3 illustrates how the multimodal clarification subdialogue of example 1 was annotated. The problem `source` of the clarification request describes the type of understanding problem which caused the need to clarify. Its attributes map to the level of 'understanding' as defined by (Clark 1996). The problem `severity` describes which type of feedback the CR-initiator requests from the other dialogue participant, i.e. asking for `confirmation` or for `elaboration/repetition`. For the antecedent we are interested in its speech act type and its arguments as shown in example 3. The reply is classified according to its information gain and the complexity of the underlying language model. These attributes reflect that a good clarification strategy for spoken dialogue systems should elicit responses which maximise information gain while minimising recognition errors. These desiderata are reflected in the values of the reply type, which are adding information (`add`), repeating an utterance (`repeat`), a y/n answer (`y/n`), or the user changes topic (`change`). Note that `change` also includes the case where the user corrects himself, i.e. changes his goal. The following example 3 shows how one clarification sub-dialogue was annotated. In this work we now concentrate on the issue of which output *modality* the wizards choose, given that the `severity` and `source` of the CR are known. In particular, we use machine learning to build predictive models of when the wizards choose a modality, based on `severity` and `source` as predictive features.[5]

[4] Note that although the wizards were trained using the classification scheme the whole setup imposed a high cognitive load on the wizards. In post debriefing sessions the wizards reported they sometimes only choose some options to make the pop-up window disappear.

[5] Since `reply-type` is not available at run-time it is not used as a predictive feature.

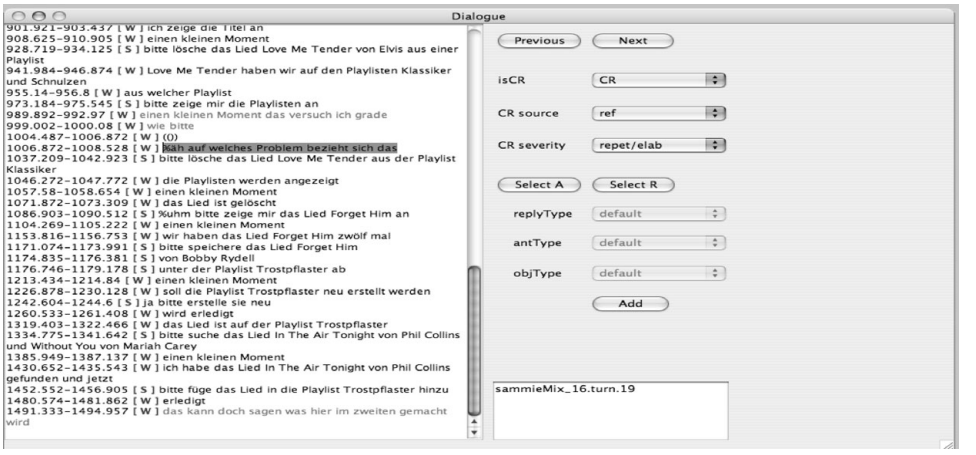



Fig. 6. The NXT-based tool for annotating CRs.

(3) **User:** Please play 'Nevermind'.
**Antecedent:** `SA-action = command`
`SA-argument = play`
**Wizard:** Does this list contain the song? *[shows list with 20 DB matches]*
**CR:** `source = reference`,
`severity = confirmation`,
`screen output = graphic-yes`
**User:** It's number 4. *[clicks on item 4]*
**Reply:** `reply-type = add`

### *3.2 Reliability*

We employed NXT-based tools (Carletta *et al.* 2003) for manual annotation. Figure 6 shows a screenshot of the CR annotation tool. It allows one to select an utterance in the left-hand side of the display by clicking on it, and then choose the attribute values from the drop-down lists on the right-hand side; one can also create and annotate relations between elements by clicking on 'Select A' (to create a CR antecedent) and 'Select R' (to create a CR reply).

The whole annotation was performed twice, by an expert and by a naïve annotator. For evaluating the reliability of the manual annotations we used the $\kappa$ coefficient (Carletta 1996). For identifying CRs we chose a cascaded approach as introduced by Carletta *et al.* (1997) to assure maximal reliability for this task with $\kappa > 0.8$. For annotating further features we only used the cases which both annotators identified as being CRs, resulting in 177 annotated CRs. The reliability of all the other features listed above is within the accepted boundaries ($0.67 < \kappa < 0.8$) (see Carggs and McGee-Wood (2005) for a discussion).

### *3.3 Statistical analysis of CRs and multimodal behaviour*

Of the 774 wizard turns 22.87 per cent were annotated as CRs. In human-human task-oriented dialogues, in contrast, the frequency of CRs is only about 5 per cent

Table 1. SAMMIE *data description*

| |
|---|
| # wizards: 6; |
| # users/sessions: 21; |
| # tasks per user: 4 (unique across wizards); |
| # dialogues: 70; |
| # turns: approx. 17,000; |
| # CRs: 177; |

(Rieser and Moore 2005), which indicates that the experimental setup is suitable to elicit clarification behaviour. Our six wizards contributed about equal proportions to the 177 data points, i.e. each wizard asked about 30 CRs. Table 1 summarises the SAMMIE data.

A $\chi^2$ test on multimodal strategy (i.e. showing a screen output or not with a CR) showed significant differences between wizards ($\chi^2(1) = 34.21, p < .000$). On the other hand, a Kruskal–Wallis test comparing user preference for the multimodal output showed no significant difference across wizards ($H(5)=10.94$, $p > .05$)[6], where mean performance ratings for the wizards' multimodal behaviour ranged from 1.67 to 3.5 on a five-point Likert scale. Observing significantly different strategies which are not significantly different in terms of user satisfaction scores, we conjecture that the wizards converged on strategies which were appropriate in certain *contexts*. To strengthen this hypothesis we split the data by wizard and and performed a Kruskal–Wallis test on multimodal behaviour per session. Only the two wizards with the lowest performance score showed no significant variation across session, whereas the wizards with the highest scores showed the most varying behaviour. In the following we test whether the observed variation is random or context-dependent, i.e. whether specific contextual features significantly contributed to the wizards different choices. We apply feature engineering methods and build a prediction model of the strategy that an *average* wizard took dependent on certain dialogue context features.

## 4 Feature extraction and feature selection/engineering

### *4.1 Context/information-state features*

A state or context in our system is a dialogue 'information state' as defined in Lemon *et al.* (2005). We divide the types of information represented in the dialogue information state into *local features* (comprising low level and dialogue features), *dialogue history features*, and *user model features*. We also defined features reflecting the application environment (e.g. `driving`). The information state features are shown in Tables 2 to 4, and further described below. All features are automatically extracted from the XML log-files (and are available at runtime in ISU-based dialogue systems). From these features we want to learn in which contexts to generate a screen output

[6] The Kruskal–Wallis test is the non-parametric equivalent to a one-way ANOVA. Since the users indicated their satisfaction on a five-point likert scale, an ANOVA which assumes normality would be invalid.

Table 2. *Contextual/information-state features: local features*

| **Local features** |
|---|
| `DBmatches`: data base matches (numeric) |
| `Deletion`: deletion rate (numeric) |
| `Delay`: delay of user reply (numeric) |
| `CR-source`: problem source (4-valued) |
| `CR-severity`: problem severity (2-valued) |
| `SA-action`: user speech act (4-valued) |
| `SA-argument`: user speech argument (9-valued) |

Table 3. *Contextual/information-state features: history features*

| **Dialogue history features** |
|---|
| `CRhist`: number of CRs (numeric) |
| `ScreenHist`: number screen outputs (numeric) |
| `DelHist`: average corruption rate (numeric) |
| `DialogueDuration`: dialogue duration (numeric) |
| `RefHist`: number of verbal user references to screen output (numeric) |
| `ClickHist`: number of click events (numeric) |

(`graphic-yes`), and when to clarify using speech only (`graphic-no`). The case that the wizard only used screen output for clarification did not occur.

### *4.1.1 Local features*

First, we extracted features present in the 'local' context of a CR, as shown in Table 2, such as the number of matches returned from the data base query (`DBmatches`), how many words were deleted by the corruption algorithm[7] (`deletion`), what problem source the wizard indicated (`source`), what problem `severity`, the previous user speech act (`SA-action`), its argument (`SA-argument`), and the delay between the last wizard utterance and the user's reply (`delay`).[8]

### *4.1.2 Dialogue history features*

The history features account for events in the whole dialogue so far, i.e. all information gathered before asking the CR, as shown in Table 3, such as the number of CRs asked (`CRhist`), how often the screen output was already used (`screenHist`), the corruption rate so far (`delHist`), the dialogue duration so far (`duration`), and whether the user reacted to the screen output, either by verbally

[7] Note that this feature is only an approximation of the ASR confidence score that we would expect in an automated dialogue system. See Rieser *et al.* (2005). for full details.
[8] We introduced the `delay` feature to handle clarifications concerning contact.

Table 4. *Contextual/information-state features: user model features*

| **User model features** |
|---|
| `ClickUser`: average number of clicks (numeric) |
| `RefUser`: average number of verbal references (numeric) |
| `DelUser`: average corruption rate for that user (numeric) |
| `ScreenUser`: average number of screens shown to that user (numeric) |
| `CRuser`: average number of CRs asked to user (numeric) |
| `Driving`: user driving (binary) |

referencing (`refHist`), e.g. using expressions such as 'It's item number 4', or by clicking (`clickHist`) as in example 3.

#### *4.1.3 User model features*

Under 'user model features' we consider features reflecting the wizards' responsiveness to the behaviour and situation of the user. Each session comprised four dialogues with one wizard. The user model features average the user's behaviour in these dialogues so far, as shown in Table 4, such as how responsive the user is towards the screen output, i.e. how often this user clicks (`clickUser`) and how frequently s/he uses verbal references (`refUser`); how often the wizard had already shown a screen output (`screenUser`) and how many CRs were already asked (`CRuser`); how much the user's speech was corrupted on average (`delUser`), i.e. an approximation of how well this user is recognised; and whether this user is currently driving or not (`driving`). This was the only driving related information available to the wizards. It would be interesting to provide more detailed information the wizards, for example, on the current driving situation or the overall driving performance of the user.

### *4.2 Discussion*

Note that all these features are generic over information-seeking dialogues where database results can be displayed on a screen; except for `driving` which only applies to hands-and-eyes-busy situations. Table 5 shows a context for the dialogue in example 3, assuming that it was the first utterance by this user. This potential feature space comprises 19 features, many of them taking numeric attributes as values. Considering our limited data set of 177 training instances we run the risk of severe data sparsity. Note that for WOZ studies the amount of available data is usually quite limited. Furthermore, we want to explore which features of this potential feature space influenced the wizards' multimodal strategies. In the next two sections we describe feature engineering techniques, namely discretising methods for dimensionality reduction and feature selection methods, which help to reduce the feature space to a subset which is most predictive of multimodal clarification. For our experiments we use implementations of discretisation and feature selection methods provided by the WEKA toolkit (Witten and Frank 2005).

Table 5. *Example: features in the context after the first turn in example 3*

```
LOCAL FEATURES
        DBmatches: 20
        Deletion: 0
        CR-source: reference resolution
        CR-severity: confirmation
        UserSpeechact: command
        Delay: 0
HISTORY FEATURES
        [CRhist, screenHist, delHist,refHist,clickHist] = 0
        duration = 10s
USER MODEL FEATURES
        [clickUser,refUser,screenUser,CRuser] = 0
        Driving = true
```

### *4.3 Discretising numeric features*

Global discretisation methods divide all continuous features into a smaller number of distinct ranges before learning starts. This has a number of advantages concerning the quality of our data for ML. First, discretisation methods reduce the size of the feature space for learning, which is especially useful when learning from small data sets. In addition, discretisation methods take feature distributions into account and help to avoid sparse data. Furthermore, most of our features are highly positively skewed. Some ML methods (such as the standard extension of the Naïve Bayes classifier to handle numeric features) assume that numeric attributes have a normal distribution. We use Proportional k-Interval (PKI) discretisation as a unsupervised method, and an entropy-based algorithm (Fayyad and Irani 1993) based on the Minimal Description Length (MDL) principle as a supervised discretisation method. PKI uses equal frequency binning, whereas MDL uses information gain (Kullback–Leibler divergence) to recursively define the best bins.

### *4.4 Feature selection*

Feature selection is the problem of selecting an optimum subset of features that are most predictive of a given outcome. The objective of selection is two-fold: improving the prediction performance of ML models and providing a better understanding of the underlying concepts that generated the data. We chose to apply forward selection for all our experiments given our large feature set, in order to not include redundant features. We use the following feature filtering methods: *correlation-based* subset evaluation (CFS) (Hall 2000) and a *decision tree* algorithm (rule-based ML) for selecting features before doing the actual learning. We also experimented with a wrapper method called *Selective Naïve Bayes* (selective Bayes), which has been shown to perform reliably well in practice (Langley and Sage 1994). We also apply a correlation-based *ranking* technique. The previous feature subset selection models model inner-feature relations, selecting subsets of predictive features at the expense

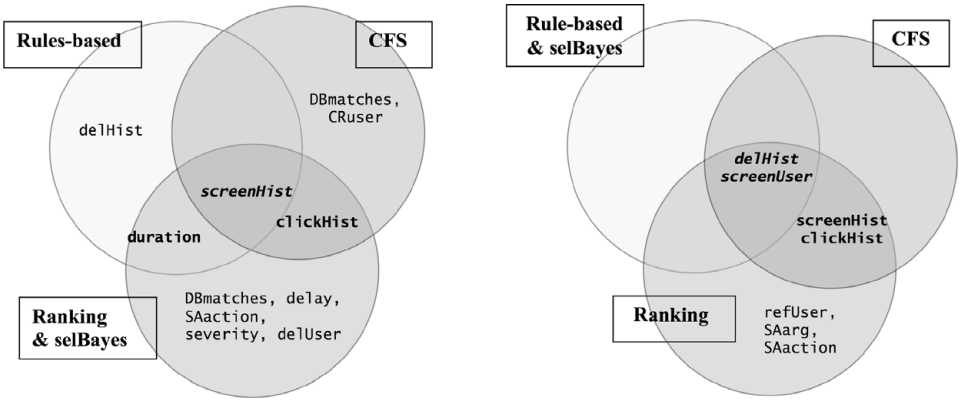



Fig. 7. Features selected to be most predictive in relation to feature engineering methods. Feature selection on PKI-discretised data (left) and on MDL-discretised data (right).

of saying less about individual feature performance itself. Ranking evaluates each feature individually.

### *4.5 Results for PKI and MDL discretisation*

Note that feature selection and discretisation interact, i.e. feature selection performs differently on PKI or MDL discretised data. MDL discretisation reduces our range of feature values dramatically. It fails to discretise 9 of 14 numeric features and bars those features from playing a role in the final decision structure because the same discretised value will be given to all instances. However, MDL discretisation cannot replace proper feature selection methods since it doesn't explicitly account for redundancy between features, nor for non-numerical features. For the other 5 features which were discretised there is a binary split around one (fairly low) threshold: `screenHist` (.5), `clickHist` (.5), `refUser` (.5), `screenUser` (.42); except for `delUser` which is only predictive around a higher threshold (8.0). The PKI algorithm discretises numeric attributes using equal frequency binning. Therefore it does not reduce the number of features used for feature selection, and different features do get selected than for MDL discretised data. Figure 7 shows the results from feature selection on PKI and MDL discretised data sets. The most frequently chosen features are `screenUser`, `screenHist`, `clickHist` and `delUser`. The features `screenUser`, `screenHist`, `clickHist` indicate that the average wizard adapts to the user behaviour. For example, if the user had been responsive to the screen output `screenUser`, `clickHist` and whether multimodal output was successfully generated before `screenHist`. Furthermore, the average wizard adapts to noise/uncertainty in the user input (`delUser`).

We now move on to explore the performance of feature engineering methods in combination with different ML algorithms (where we treat feature optimisation as an integral part of the training process).

## 5 Using supervised learning to explore human multimodal clarification strategies

We use a wide range of different multivariate classifiers which reflect our hypothesis that a CR decision is based on various features in the context, and compare them against a simple baseline strategy, reflecting deterministic contextual behaviour. All experiments are carried out using 10-fold cross-validation. We take an approach similar to (Daelemans *et al.* 2003) where parameters of the classifiers are optimised with respect to feature selection.

### *5.1 Baseline*

The simplest baseline we can consider is to always predict the majority class in the data, in our case `graphic-no` (74.1 per cent ). This yields a 62.9 per cent weighted f-score (described below). This baseline reflects a deterministic wizard strategy of never showing a screen output.

### *5.2 Machine learners*

For learning we experiment with five different types of supervised classifiers: Rule Induction, Decision Trees, Naïve Bayes, Bayesian Networks and Maximum Entropy.[9] We chose Naïve Bayes as a joint (generative) probabilistic model, using the WEKA implementation of John and Langley (1995)'s classifier; Bayesian Networks as a graphical generative model, again using the WEKA implementation; and we chose Maximum Entropy as a discriminative (conditional) model, using a Maximum Entropy toolkit (Le 2003). As a rule induction algorithm we used JRIP, the WEKA implementation of Cohen (1995)'s 'Repeated Incremental Pruning to Produce Error Reduction' (RIPPER). And for decision trees we used the J4.8 classifier (WEKA's implementation of the C4.5 system (Quinlan 1993)).

### *5.3 Results: comparing performance of different learners and feature engineering methods*

We experimented using these different classifiers on raw data on MDL and PKI discretised data and on discretised data using the different feature selection algorithms. We report on two measures: accuracy and weighted f-score. The weighted f-score is the weighted sum (by class frequency in the data) of the f-scores of the individual classes (25.9 per cent `graphic-yes`, 74.1 per cent `graphic-no`).[10]

In Table 6 we see fairly stable high performance for Bayesian models with MDL feature selection. However, the best performing model is Naïve Bayes using wrapper methods (selective Bayes) for feature selection and PKI discretisation. This model

[9] In a pre-study we also experimented with k-nearest neighbours, as an instance-based classifier, which did not show significant results.

[10] The following results are an update of (Rieser and Lemon 2006). The results presented here are also based on more accurate manual annotations whereas in (Rieser and Lemon 2006) the results were based on less reliable automatically annotated data.

Table 6. *Weighted f-scores and accuracy for learned models*

| f.engin./ (wf-score /acc.(%)) | Majority baseline | Rule Induction | Decision tree | Maximum entropy | Naïve bayes | Bayesian network |
|---|---|---|---|---|---|---|
| raw | **62.9/74.0** | 80.4/81.4 | 80.7/78.8 | 78.1/79.0 | 80.8/80.5 | 79.9/77.7 |
| PKI | 62.9/74.0 | 76.4/75.9 | 79.1/79.8 | 77.6/78.5 | 82.5/83.0 | 85.0/83.2 |
| PKI-CFS | 62.9/74.0 | 76.5/76.6 | 79.2/79.6 | 82.0/82.4 | 81.7/83.4 | 83.7/84.4 |
| PKI-rule | 62.9/74.0 | 78.8/82.2 | 76.9/79.4 | 81.5/83.2 | 80.3/79.8 | 80.3/79.8 |
| PKI-selB | 62.9/74.0 | 78.0/78.8 | 79.5/81.6 | 82.5/82.2 | **88.5/87.8** | **87.5/87.6** |
| PKI-rank | 62.9/74.0 | 79.3/79.4 | 78.3/80.9 | 83.1/83.2 | 83.8/84.4 | 85.0/84.3 |
| MDL | 62.9/74.0 | 83.2/81.6 | **84.3/84.7** | 79.6/80.3 | 81.6/79.6 | 80.7/78.8 |
| MDL-CFS | 62.9/74.0 | 83.5/83.7 | 84.0/84.2 | 84.2/84.3 | 84.0/84.1 | 84.0/84.1 |
| MDL-rule | 62.9/74.0 | 84.0/84.1 | 84.0/84.1 | 84.2/84.4 | 84.0/84.1 | 84.0/84.1 |
| MDL-selB | 62.9/74.0 | **84.0/84.1** | 84.0/84.1 | 83.6/83.5 | 84.0/84.1 | 84.0/84.1 |
| MDL-rank | 62.9/74.0 | 82.2/81.9 | 84.0/84.1 | 78.9/78.2 | 83.5/83.6 | 81.5/81.3 |

achieves a wf-score of 88.55 per cent , which is a 25.6 per cent improvement over the baseline (see Table 6, bold print).

We observe main effects for discretisation method, feature selection method and ML algorithms. We also find significant interactions between discretisation method and ML algorithms, as well as between discretisation and feature selection. We now separately explore the models and feature engineering techniques and their impact on the prediction accuracy for each trial/cross-validation. We compare the group means for models, discretisation and feature selection methods using a Kruskal–Wallis test with Mann–Whitney tests as a post-hoc procedure (using Bonferroni correction for multiple comparisons), we obtained the following results:

- All ML algorithms are significantly better than the baseline. There is no significant difference in the performance of Decision tree, Maximum entropy, Naïve Bayes and Bayesian network classifiers.
- For discretisation methods we found that the classifiers were performing significantly better on MDL discretised data than on PKI or continuous data. MDL being significantly better than non-discretised data indicates that all wizards behaved as though using 'thresholds' to make their decisions. Supervised MDL being better than unsupervised PKI supports the hypothesis that decisions were context dependent, as MDL considers underlying features for discretisation.
- All feature selection methods (except for CFS) lead to better performance than using all of the features. Selective Bayes and rule-based selection performed significantly better than CFS. Selective Bayes, rule-based learning and subset-overlap showed no significant differences. These results show that wizards behaved as though specific features were important (but they suggest that the inner-feature relations used by CFS are less important).

```
IF screenHist > .5 AND deletionUser <8 THEN graphic=yes;
ELSE graphic=no;
```

Fig. 8. Reformulation of the rules learned by JRIP.

#### *5.3.1 Discussion of results*

These experimental results support several main points. First, the results indicate that we can learn a good prediction model from our data. We conclude that our six wizards did not behave arbitrarily, but selected their strategy according to various contextual features. By separating out the individual contributions of models and feature engineering techniques, we have shown that wizard behaviour is based on multiple features in the dialogue. The best results were achieved by Decision Tree, Maximum Entropy, classifiers on MDL discretised data, and Naïve Bayes and Bayesian Network classifiers on PKI discretised data. This confirms the interaction between discretisation method and ML algorithms mentioned above. All the best performing models use selective Bayes as feature selection technique, which uses the richest feature space including the features `DBmatches`, `delay`, `duration`, `SA-action`, `severity`, `screenHist`, `clickHist`, `delUser`. This illustrates that models learned from small data sets can achieve high accuracy, even using a large set of features, if feature selection is applied beforehand. The overall best performing model is selective Bayes on PKI disretised data.

### *5.4 Interpretation of the learned strategies*

For interpreting the learned strategies we will discuss the results of rule induction and decision trees since they are the easiest to interpret (and also to implement in standard rule-based dialogue systems). For both we explain the models producing the best results, which is MDL in combination with selective Bayes for rule induction, and MDL without any feature selection for decision trees (see Table 6, bold print).

#### *5.4.1 Rule induction*

Figure 8 shows a reformulation of the rules from which the learned classifier is constructed. The feature `screenUser` plays a central role. These rules (in combination with the low thresholds) say that if you have already shown a screen output in this dialogue before (i.e. `screenHist>.5`), then do so again if the acoustic understanding is quite *reliable* (i.e. `deletionUser<8`). Otherwise don't show screen output when asking a clarification.

This strategy recommends not showing a multimodal output if the uncertainty introduced by speech recognition rises. This contradicts the observations by (Oviatt 2002; Oviatt *et al.* 2004) that multimodal generation should be used in environments with poor speech recognition. We believe that this result is explained by the observation that our wizards clearly behaved sub-optimally in these situations, as humans are normally not confronted with simulated speech recognition errors

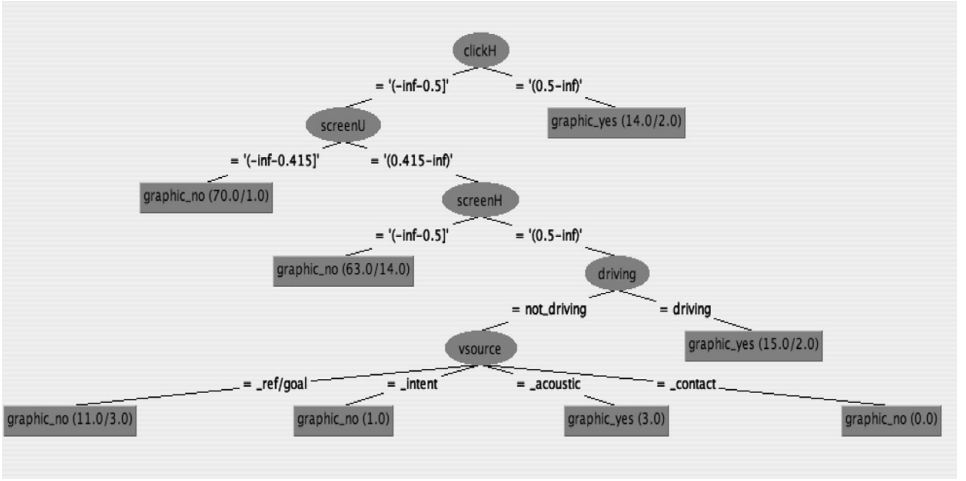



Fig. 9. Five-rule tree from J4.8 ('inf' $= \infty$).

(and even deleted words) and are not experts in asking multimodal CRs in this domain.

### 5.4.2 Decision tree

Figure 9 shows the decision tree learned by the classifier J4.8 on MDL discretised data. The eight rules contained in the tree rely on the features `driving` and `source` which were not previously chosen by feature selection algorithms. The rules constructed by the first three nodes are saying that, if (in this dialogue) the user has chosen an item by clicking on it, then always generate a graphic when asking a CR. Also don't show a graphic if the user has not clicked yet, and if the user has never used the multimodal interface in any previous dialogue (`screenUser`) and hasn't done so in the current dialogue (`screenHist`). Do show a graphic if the user has used the graphical output in any previous dialogue, and if the user is driving. This wizard strategy disagrees with findings by (Salmen 2002) that showing long lists while driving increases the cognitive load for the user.

## 5.5 Discussion

The strategies learned by the classifiers from wizard behaviours contradict some of the findings of user studies on how multimodality should be used to gain optimal results. However, we now know that our wizards did not behave optimally in this situation, so the issue for future work is whether we can use different learning techniques to move from observed wizard behavior to optimal behaviour.

For this we are currently using RL methods. For learning a strategy which varies in context but adapts in more subtle ways (e.g. to the user model), we would need to explore many more strategies through interactions with users to find an optimal one. One way to reduce costs for building such an optimised strategy is to apply RL with simulated users. RL has been shown to lead to dialogue strategies which are

better than the human strategies present in the original data (Henderson, Lemon and Georgila 2008; Lemon, Georgila and Henderson 2006). In current work we also explore the use of the selected features to define the state-space for RL. We can show that strategies optimised with RL significantly outperform strategies such as the above, which mimick human wizards' behaviour (Rieser and Lemon 2008).

## 6 Conclusion and future Work

We have shown that humans use context-dependent strategies for asking multimodal clarification requests, and we learned such strategies from WOZ data. Only the two wizards with the lowest performance scores showed no significant variation across sessions, leading us to hypothesise that the better wizards converged on a context-dependent strategy. We were able to discover a runtime dialogue context representation based on which all wizards behaved uniformly, using feature discretisation methods and feature selection methods on dialogue context features. Based on these features we were able to predict how an 'average' wizard would behave in that context with an accuracy of 87.88 per cent (wf-score of 88.55 per cent , which is a 25.6 per cent improvement over the majority baseline). We explained and interpreted the learned strategies and showed that they can be implemented in rule-based dialogue systems based on domain independent features. We also showed that feature engineering is essential for achieving significant performance gains when using large feature spaces with the small data sets which are typical of dialogue WOZ studies. By interpreting the learned strategies we found them to be suboptimal. In our ongoing research, RL is applied to optimise strategies and has been shown to lead to dialogue strategies which are better than those present in the original data (Rieser and Lemon 2008).

## Acknowledgements

The authors would like thank Ivana Kruijff-Korbayová, Nate Blaylock, Ciprian Gerstenberger, Tilman Becker, Michael Kaißer, Peter Poller, Jan Schehl for setting up the WOZ experiments together. This research has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 216594 (CLASSIC project: www.classic-project.org), and by EPSRC grant number EP/E019501/1.

## References

Carletta, J. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistic* **2**(22): 249–254.

Carletta, J., Evert, S., Heid, U., Kilgour, J., Robertson, J., and Voormann, H. 2003. The NITE XML Toolkit: flexible annotation for multi-modal language data. *Behavior Research Methods, Instruments, and Computers, special issue on Measuring Behavior* **35**(3): 353–363.

Carletta, J., Isard, A., Isard, S., Kowtko, J. C., Doherty-Sneddon, G., and Anderson, A. H. 1997. The reliability of a dialogue structure coding scheme. *Computational Linguistics* **1**(23): 13–31.

Clark, H. 1996. *Using Language*. Cambridge University Press, Cambridge.

Cohen, W. W. 1995. Fast effective rule induction. In *Proceedings of the 12th ICML-95*.

Craggs, R., and McGee-Wood, M. 2005. Evaluating discourse and dialogue coding schemes. *Computational Linguistics* **31**(3): 289–296.

Daelemans, W., Hoste, V., De Meulder, F., and Naudts, B. 2003. Combined optimization of feature selection and algorithm parameter interaction in machine learning of language. In *Proceedings of the 14th ECML-03*.

Fayyad, U., and Irani, K. 1993. Multi-interval discretization of continuous valued attributes for classification learning. In *Proc. IJCAI-93*.

Hall, M. 2000. Correlation-based feature selection for discrete and numeric class machine learning. In *Proc. 17th Int Conf. on Machine Learning*.

Henderson, J., Lemon, O., and Georgila, K. 2008. Hybrid reinforcement/supervised learning of dialogue policies from fixed datasets. *Computational Linguistics* **34**(4): 487–513.

John, G., and Langley, P. 1995. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the 11th UAI-95*. Morgan Kaufmann.

Kruijff-Korbayová, I., Becker, T., Blaylock, N., Gerstenberger, C., Kaisser, M., Poller, P., Rieser, V., and Schehl, J. 2006a. The SAMMIE corpus of multimodal dialogues with an MP3 player. In *Proceedings the 5th International Conference on Language Resources and Evaluation (LREC)*.

Kruijff-Korbayová, I., Blaylock, N., Gerstenberger, C., Rieser, V., Becker, T., Kaisser, M., Poller, P., and Schehl, J. 2005. An experiment setup for collecting data for adaptive output planning in a multimodal dialogue system. In *10th European Workshop on NLG*.

Kruijff-Korbayová, I., Rieser, V., Gerstenberger, C., Schehl, J., and Becker, T. 2006b. The Sammie multimodal dialogue corpus meets the Nite XML Toolkit. In *Proceedings of the Fifth Workshop on multi-dimensional Markup in Natural Language Processing*.

Langley, P., and Sage, S. 1994. Induction of selective Bayesian classifiers. In *Proceedings of the 10th UAI-94*.

Le, Z. 2003. *Maximum Entropy Modeling Toolkit for Python and C++*. `homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html`.

Lemon, O., Georgila, K., and Henderson, J. 2006. Evaluating effectiveness and portability of reinforcement learned dialogue strategies with real users: the TALK TownInfo evaluation. In *IEEE/ACL Spoken Language Technology*.

Lemon, O., Georgila, K., Henderson, J., Gabsdil, M., Meza-Ruiz, I., and Young, S. 2005. Deliverable D4.1: integration of learning and adaptivity with the ISU approach. Technical report, TALK Project, `www.talk-project.org`.

Mattes, S. 2003. The lane-change-task as a tool for driver distraction evaluation. In *Proc. of IGfA*.

Oviatt, S. 2002. Breaking the robustness barrier: recent progress on the design of robust multimodal systems. In *Advances in Computers*, vol. 56, Academic Press, London.

Oviatt, S., Coulston, R., and Lunsford, R. 2004. When do we interact Multimodally? Cognitive load and multimodal communication patterns. In *Proceedings of the 6th ICMI-04*.

Purver, M., Ginzburg, J., and Healey, P. 2003. On the means for clarification in dialogue. In R. Smith, and J. van Kuppevelt (eds.), *Current and New Directions in Discourse and Dialogue*, Dordrecht, The Netherlands.

Quinlan, R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo.

Rieser, V. 2008. *Bootstrapping Reinforcement Learning-based Dialogue Strategies from Wizard-of-Oz data*. Ph.D. thesis, Saarbruecken Dissertations in Computational Linguistics and Language Technology, Vol 28.

Rieser, V., Kruijff-Korbayová, I., and Lemon, O. 2005. A corpus collection and annotation framework for learning multimodal clarification strategies. In *Proceedings of the 6th SIGdial Workshop on Discourse and Dialogue*.

Rieser, V., and Lemon, O. 2006. Utilising machine learning to explore human multimodal clarification strategies. In *Proceedings of the 44rd Annual Meeting of the Association for Computational Linguistics, COLING/ACL*.

Rieser, V., and Lemon, O. 2008. Learning effective multimodal dialogue strategies from Wizard-of-Oz data: bootstrapping and evaluation. In *Proceedings of ACL*.

Rieser, V., and Lemon, O. 2009. Natural language generation as planning under uncertainty for spoken dialogue system. In *Proceedings of EACL*.

Rieser, V., and Moore, J. 2005. Implications for generating clarification requests in task-oriented dialogues. In *Proceedings of the 43rd ACL*.

Rodriguez, K., and Schlangen, D. 2004. Form, intonation and function of clarification requests in German task-orientaded spoken dialogues. In *Proceedings of the Eighth Workshop on Formal Semantics and Dialogue*.

Salmen, A. 2002. *Multimodale Menüausgabe im Fahrzeug (Multimodal Menu-based Interaction in the Vehicle)*. Ph.D. thesis, University of Regensburg.

Schlangen, D., and Fernandez, R. 2007. Speaking through a noisy channel: experiments on inducing clarification behaviour in human–human dialogue. In *Interspeech*.

Skantze, G. 2005. Exploring human error recovery strategies: implications for spoken dialogue systems. *Speech Communication* **43**(3): 325–341.

Stuttle, M. N., Williams, J. D., and Young, S. 2004. A framework for dialogue data collection with a simulated ASR Channel. In *ICSLP*.

Walker, M., Whittaker, S., Stent, A., Maloor, P., Moore, J., Johnston, M., and Vasireddy, G. 2004. User tailored generation in the MATCH multimodal dialogue system. *Cognitive Science*, **28**: 811–840.

Winterboer, A., Hu, J., Moore, J. D., and Nass, C. 2007. The influence of user tailoring and cognitive load on user performance in spoken dialogue systems. in *Proc. ICSLP*.

Witten, I. H., and Frank, E 2005. *Data Mining: Practical Machine Learning Tools and Techniques (2nd Edition)*. Morgan Kaufmann, San Francisco.